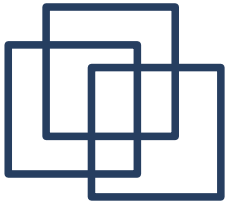


Кафедральный практикум 5 семестр

Часть 1. Введение в функциональное программирование

<http://sp.cmc.msu.ru/~kornyxin/fp/slides/part1-1.pdf>



План

Часть 1.

1) Организационные вопросы

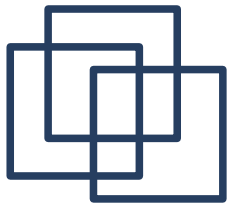
2) Функциональный стиль программирования

Часть 2. Язык программирования Scheme.

Часть 3. Функции высшего порядка.

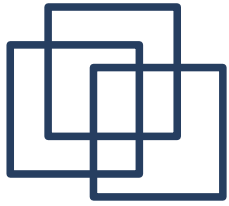
«Векторное» мышление.

Часть 4. Теоретический фундамент ФП.



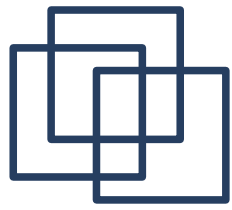
Орг.вопросы

- Что предстоит сдавать:
 - Домашние работы (ejudge)
 - Практические задания (виртуальная машина и генетическое программирование)
 - Контрольные работы в классе
- Материалы: <http://sp.cmc.msu.ru/~kornykhin>
- Вопросы, пожелания, предложения:
 - kornevgen@cmc.msu.ru (Корныхин Евгений)
 - amonakov@ispras.ru (Монаков Александр)



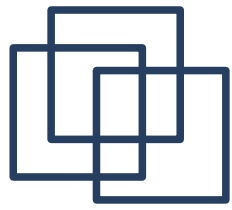
Инструменты

- Как называется, откуда скачать, как установить, как пользоваться, что надо делать дополнительно, чтобы задача прошла в ejudge, *будет рассказано и показано на следующем семинаре!*



«Expression only» стиль

- примеры statement'ов :
 - присваивание (всех видов, в том числе ++)
 - `if (...) { ... } else { ... }`
 - `while (...) { ... }`
 - `for (...; ...; ...) { ... }`
 - `return ...;`
 - «expression statement» (пример: «`x+1;`»)
- примеры expression'в :
 - арифметика, булевские выражения и т.п.
 - вызов функции
 - оператор ? :



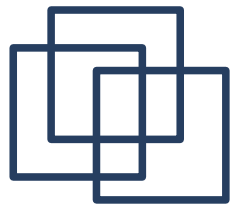
«Expression only» стиль

- «Expression only» - значит, без statement'ов (**кроме return**).
- Тело функции — return с выражением.

```
int f(int x)
{
    int y = x;
    y++;
    return y+x;
}
```



```
int f(int x)
{
    return 2*x+1;
}
```



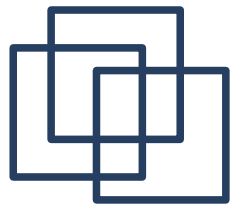
«Expression only» стиль

- «Expression only» - значит, без statement'ов (**кроме return**).
- Тело функции — return с выражением.

```
int max(int x,  
        int y)  
{  
    int m = x;  
    if (y > x)  
        m = y;  
    return m;  
}
```



```
int max(int x,  
        int y)  
{  
    return y > x ?  
        y : x;  
}
```



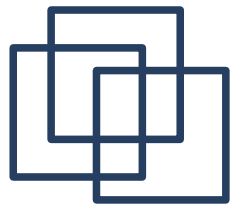
«Expression only» стиль

- Вместо выражения ? : будем использовать в том же смысле if-then-else (else обязательно)

```
int max(int x,  
        int y)  
{  
    int m = x;  
    if (y > x)  
        m = y;  
    return m;  
}
```



```
int max(int x,  
        int y)  
{  
    if (y > x)  
        return y;  
    else  
        return x;  
}
```

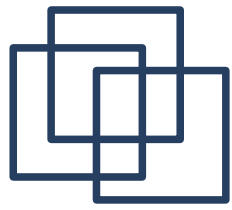



«Expression only» стиль

```
long f(int n)
{
    long m = 1;
    for(int i = 1;
        i < n; i++)
        m *= i+1;
    return m;
}
```



```
long f(int n)
{
    if (n == 0)
        return 1;
    else
        return n*f(n-1);
}
```



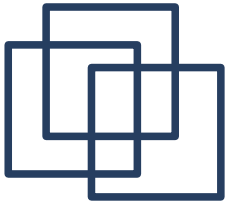
«Expression only» стиль

- Возможны константные локальные переменные (как подвыражения).

```
int g(int x);
int f(int x)
{
    int y = g(x);
    y++;
    int z = 2*g(x);
    return y+z;
}
```



```
int g(int x);
int f(int x)
{
    int t = g(x);
    return 3*t+1;
}
```

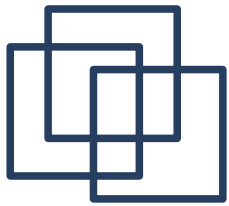


Пример

- Переписать в стиле «только из выражений»

```
int gcd(int a, int b)
{
    while (a > 0 && b > 0) {
        if (a > b) a = a - b;
        else b = b - a;
    }

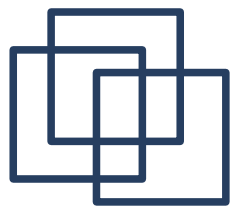
    if (a == 0) return b;
    else return a;
}
```



Другое решение

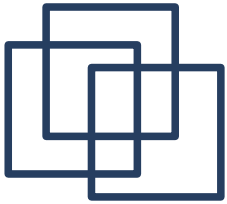
```
int max(list<int> xs);
int common(list<int> xs, list<int> ys);
list<int> divisors(int x);

int gcd(int a, int b)
{
    return max( common ( divisors(a),
                        divisors(b) ) );
}
```



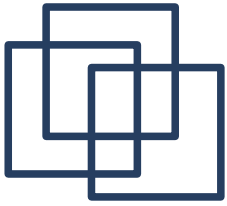
Обработка деревьев

- Вы часто встречались со стилем `expression only` для обработки деревьев
 - рекурсивные функции обхода
 - $f(\text{root}) = g(f(\text{left}), f(\text{right}), \text{root})$
- Примеры: вычисление глубины дерева, поиск элемента в дерева, вставка в дерево (убедитесь в этом!)
- Для любителей «подумать»: почему так писали эти функции? как писать их иначе?



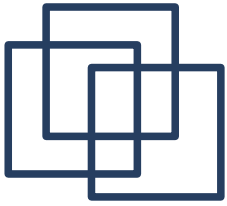
На C++

- Стиль «expression only» на C++ (без if):
 - все формальные параметры функций — константные
 - отсутствие глобальных переменных
 - все локальные переменные — константные
 - нет функций, возвращающих void
 - нет statement'ов, за исключением return
 - только один return в функции
 - всё остальное отследит компилятор
- Доп. опции компилятора g++: -Wall -Werr



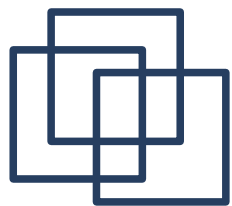
На C++

- Стиль «expression only» на C++ (**c if**)
дополнительно:
 - нет statement'ов за исключением return **и if-then-else**
 - ~~только один return в функции~~
 - if обязательно содержит else
 - все ветви каждого if заканчиваются return
 - нет «мёртвого» кода
 - всё остальное отследит компилятор
- Доп. опции компилятора g++: -Wall -Werr



Пример

- Переписать в стиле «только из выражений» на этом подмножестве C++ только что написанный алгоритм Евклида.
- Оттранслировать написанную программу.
- Написать main, который запускает алгоритм Евклида на некоторых числах и выводит результат.
- Запустить main, исправить обнаруженные ошибки, если таковые будут иметь место.



Домашнее задание

- 1) (сделаем его сейчас!) Какую программу/задачу все знают?
- 2) Переписать в стиле «из одних выражений» несколько Си-шных функций
 - <http://sp.cmc.msu.ru/~kornyxhin> → hw #1.1
- 3) Реализовать в стиле «из одних выражений» несколько функций на C++
 - <http://sp.cmc.msu.ru/~kornyxhin> → hw #1.2
- 4) Решения оформлять в виде текстовых файлов и присылать мне на почту kornevgen@cmc.msu.ru
- 5) Подумать, что дает такой стиль написания программ для качества программ, дает ли новую парадигму программирования